## REMARKS

Claims 1-27, 37-60 are pending, with claims 1, 10, 17, 37, 45, and 51 being independent. Claims 28-36 and 61-69 have been cancelled without prejudice. Claims 1, 10, 17, 37, 45, and 51 have been amended. Support for the amendments may be found, for example, on pages 8-9 and Figs. 5-7 of the application. No new subject matter has been added.

## 102 Rejections

Claims 1-6, 8-13, 16-35, 37-42, 44-48, 51-68 stand rejected under 35 U.S.C. 102(b) as being anticipated by Stevens (TCP/IP Illustrated Volume 1.)

As amended, claim 1 recites, among other features, "transmitting a request for a data transfer session from a client device to a host, the request identifying a plurality of data objects to be transferred between the client device and the host" and "in response to the received request, transmitting from the host to the client device the plurality of identifiers for data objects, wherein each identifier is assigned by the host and corresponds to a different one of the data objects to be transferred" Stevens fails to teach or suggest at least the above limitations.

Stevens discloses the standard TCP/IP protocol session establishment. In the standard TCP/IP transaction, the client and the server exchange SYN messages to open a session, as shown in the figure below [Stevens, page 232, Fig. 18.3]:

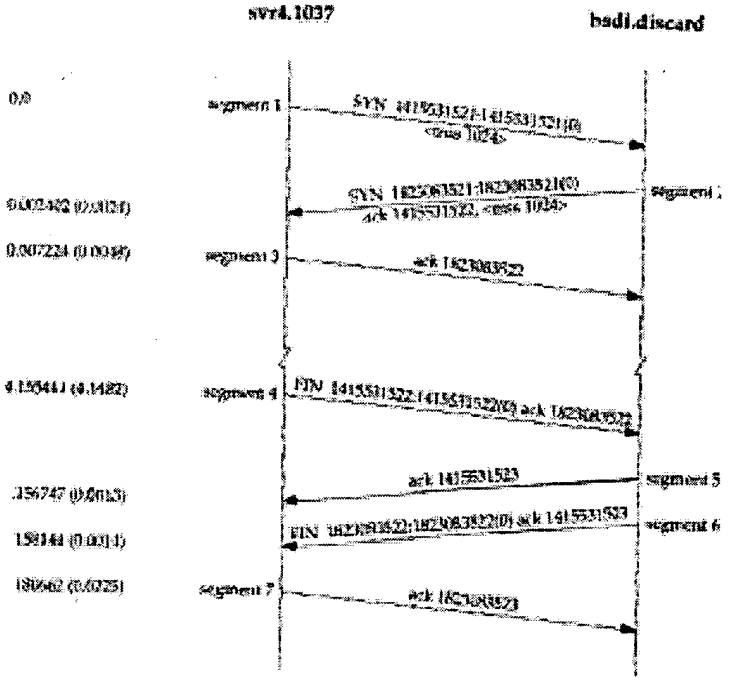


Figure 18.3 Time line of connection establishment and connection termination.

As Stevens explains, a three-way handshake between a client and a server involves an exchange of SYN and ACK messages containing *sequence numbers.* [Stevens, p. 231, "Connection Establishment Protocol" section.] Stevens explains that the client always chooses sequence numbers for its own SYNs and ACKs, while the server always selects sequence numbers for its messages: [Stevens, p. 232-233.]:

> How are the sequence numbers chosen? In 4.4BSD (and most Berkeley-derived implementations) when the system is initialized the initial send sequence number is initialized to 1. This practice violates the Host Requirements RFC. (A comment in the code acknowledges that this is wrong.) This variable is then incremented by 64,000 every half-second, and will cycle back to 0 about every 9.5 hours. (This corresponds to a counter that is incremented every 8

The significance of the above passage is that in TCP/IP, neither the server nor the client ever choose sequence numbers for the other side. Rather, each side assigns its own sequence number to the data frames it transmits to the other party. Each side is also responsible for incrementing its own the sequence numbers for the subsequent frames. [See Fig. 18.13.]

In contrast, claim 1 recites "transmitting from the host to the client device the plurality of identifiers for data objects, wherein each identifier is assigned by the host and corresponds to a different one of the data objects to be transferred." Thus, the host assigns an identifier to each data object specified in the request sent from the client device to the host. Standard TCP/IP neither describes nor suggests such a data flow.

Moreover, TCP/IP does not have a concept of "data objects." Rather, TCP itself is oblivious to the boundaries between objects/files (or their contents), as it simply provides a relievable "byte stream service" between two sides [Stevens, p. 224]:

> A stream of 8-bit bytes is exchanged across the TCP connection between the two applications. There are no record markers automatically inserted by TCP. This is what we called a *byte stream service*. If the application on one end writes 10 bytes, followed by a write of 20 bytes, followed by a write of 50 bytes, the application at the other end of the connection cannot tell what size the individual writes were. The other end may read the 80 bytes in four reads of 20 bytes at a time. One end puts a stream of bytes into TCP and the same, identical stream of bytes appears at the other end.
>
> Also, TCP does not interpret the contents of the bytes at all. TCP has no idea if the data bytes being exchanged are binary data, ASCII characters, EBCDIC characters, or whatever. The interpretation of this byte stream is up to the applications on each end of the connection.

The TCP buffer is read by higher-level protocols, which extract data objects from the stream. TCP, on the other hand, simply ensures that all bytes of data in the buffer is properly transferred between a client and a server.[Id.] Therefore, TCP/IP control packets do not reference any "data object" information, such as identifiers for data objects. For example, all TCP messages reference sequence numbers, which do not identify data objects but rather indicate the number of bytes transmitted by the sender.

Finally, with respect to claim 10, the server in TCP never transmits "to a client device a plurality of identifiers and routings of one or more handling processes," as recited in claim 10. As explained above, TCP is oblivious about the application layer processes and how they handle their own objects, so TCP/IP control packets do not include any information related to handling processes for the data objects.

For at least the above reasons, Applicants respectfully request reconsideration and withdrawal of the rejection of the independent claim 1 and its dependent claims. Independent claims 10, 17, 37, 45, and 51 recite similar transactions between the client device and the host. Claim 10 recites "transmitting, from a client device to a host, a request for a data transfer session, the request identifying a plurality of data objects to be transferred between the client device and the host" and "in response to the received request, transmitting to a client device a plurality of identifiers and routings of one or more handling processes, wherein each identifier is assigned by the host and corresponds to one of the data objects identified in the request." Claim 17 recites "receiving from a client device a request a data transfer session, the request identifying a plurality of data objects to be transferred between the client device and the host" and "sending to the client device a frame defining a session protocol that assigns an identifier to each data object identified in the request, wherein each identifier is assigned by the host." Claims 37, 45, and 51 are the computer readable medium claims corresponding to method claims 1, 10, and 17. Therefore, claims 10, 17, 37, 45, and 51 are allowable over Stevens and Kent at least for the same reasons explained with respect to claim 1.

Applicants thus respectfully request reconsideration and withdrawal of the rejection of the independent claims 1, 10, 17, 37, 45, and 51 and their dependent claims.

**103 Rejections**

Claims 7, 14, 15, 36, 43, 49, 50, and 69 stand rejected under 35 U.S.C. 102(b) as being obvious over Stevens in view of the Official Notice.

With respect to these claims, Applicants respectfully request reconsideration and withdrawal of the rejection because Official Notice does not remedy the failure of Stevens to

describe or suggest the subject matter of the independent claims and, notably, is not relied upon in the Office Action as describing the features missing from those references.

Applicant submits that all claims are in condition for allowance.

The $130 fee for a One-Month Extension of Time .is being paid concurrently herewith on the Electronic Filing System (EFS) by way of Deposit Account authorization. Please apply any other charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date:3/17/09

/Dmitry Brant/
Dmitry Brant
Reg. No. 59,333

Fish & Richardson P.C.
1425 K Street, N.W.
11th Floor
Washington, DC 20005-3500
Telephone: (202) 783-5070
Facsimile: (877) 769-7945